# Should we require content in mandatory elements?



## Background

In the context of testing the draft schema for EAC-CPF 2.0, specifically with regard to the element <recordId>, the question came up whether we should require mandatory elements to have content rather than simply being there, but being empty. During the EAC team meeting on 8 January, the additional question came up whether a mandatory element could be left empty, if the main information to be transferred by this element could also be given in an attribute. The extracts below show the minimum of a current EAC-CPF (1.0) file, the minimum of a new EAC-CPF 2.0 file, as well as the minimum of a current EAD3 (deprecated) file for comparison and to identify the elements that might be affected.

## Position of the EAC-CPF team (8 January 2021)

For the case of <recordId>, the agreement was to keep with the current mechanism in EAC-CPF 1.0 that requires <recordId> to have content, but to open up the type of content that could be in <recordId> to being any kind of text, i.e. allowing for blanks and other special characters as EAD3 currently does. There furthermore was the understanding that we would not review this for all elements, but would concentrate on those elements that currently have to be there for a minimum version of an EAS instance to validate.

## Current EAC-CPF (1.0)

The following extract from a current EAC-CPF (1.0) file is included mainly here for comparison with the draft of EAC-CPF 2.0, though apart from the mentioned changes of element to attribute, there have not been any changes regarding the mandatory elements for a minimum valid file.

```
<eac-cpf xmlns="urn:isbn:1-931666-33-4"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:isbn:1-931666-33-4 file:/Users/user/Documents/APEF/XML/eac-cpf.xsd">
   <control>
      <recordId>1</recordId>
      <!-- Needs to be there and needs to have content. -->
      <maintenanceStatus>new</maintenanceStatus>
      <!-- Needs to be there and needs to have content. There is a predefined list of values. Note: in EAC-CPF
      2.0, this is transformed to being a mandatory attribute of <control>. -->
```

```
      <maintenanceAgency>
         <agencyName></agencyName>
         <!-- Needs to be there, but can be left empty. -->
      </maintenanceAgency>
      <maintenanceHistory>
         <maintenanceEvent>
            <eventType>created</eventType>
            <!-- Needs to be there and needs to have content. There is a predefined list of values. Note: in
            EAC-CPF  2.0, this is transformed to being a mandatory attribute of <maintenanceEvent>. -->
            <eventDateTime></eventDateTime>
            <!-- Needs to be there, but can be left empty. -->
            <agentType>human</agentType>
            <!-- Needs to be there and needs to have content. There is a predefined list of values. Note: in
            EAC-CPF  2.0, this is transformed to being a mandatory attribute of <agent>. -->
            <agent></agent>
            <!-- Needs to be there, but can be left empty. -->
         </maintenanceEvent>
      </maintenanceHistory>
   </control>
   <cpfDescription>
   <!-- Either <cpfDescription> or <multipleIdentities> needs to be present with the latter than requiring at least
   two <cpfDescription>-s in itself. -->
      <identity>
         <entityType>person</entityType>
         <!-- Needs to be there and needs to have content. There is a predefined list of values. Note: in EAC-CPF
         2.0, this is transferred to the mandatory attribute @value, with <entityType> not having the option of
         [text] anymore. -->
         <nameEntry>
          <!-- Either <nameEntry> or <nameEntryParallel> needs to be present with the latter then requiring at
          least two <nameEntry>-s in itself. -->
            <part></part>
            <!-- Needs to be there, but can be left empty. -->
         </nameEntry>
      </identity>
   </cpfDescription>
</eac-cpf>
```

## Current draft for EAC-CPF 2.0

```
<eac xmlns="https://archivists.org/ns/eac/v2"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="https://archivists.org/ns/eac/v2
https://raw.githubusercontent.com/SAA-SDT/eac-cpf-schema/development/xml-schemas/eac-cpf/cpf.xsd">
   <control maintenanceStatus="new">
      <recordId>1</recordId>
      <!-- Needs to be there and needs to have content. -->
      <maintenanceAgency>
         <agencyName></agencyName>
         <!-- Needs to be there, but can be left empty. In EAC-CPF 2.0, <agencyName> allows for @valueURI, for
         which there might be the question if we would allow the URI to replace the literal, i.e. having something
         like <agencyName valueURI=”[some URI]”/>, or rather highly recommending to have a literal
         nonetheless, e.g. for the purpose of display and activation of clickable links, i.e. having something like
        <agencyName valueURI=”[some URI]”>[some name]</agencyName>. -->
      </maintenanceAgency>
      <maintenanceHistory>
         <maintenanceEvent maintenanceEventType="created">
            <eventDateTime></eventDateTime>
```

```
            <!-- Needs to be there, but can be left empty. Similar to the case of <agencyName> and @valueURI, it
            might - in this case - be an option to allow for the attribute @standardDateTime to replace the literal in
            the element. While the context of this specific piece of information might lean more towards the
            “attribute-only” option, there still might be use for highly recommending to also have a literal, e.g. to
            show the date of the last update to an EAS instance. Although, this could more easily be referred from
            the value of @standardDateTime than in the case of @valueURI. -->
            <agent agentType="human"></agent>
            <!-- Needs to be there, but can be left empty. In EAC-CPF 2.0, <agent> allows for @valueURI, for
            which there might be the question if we would allow the URI to replace the literal, i.e. having something
            like <agent agentType=”human” valueURI=”[some URI]”/>, or rather highly recommending to have a
            literal nonetheless,  e.g. for the purpose of display and activation of clickable links, i.e. having
            something like <agent agentType=”human” valueURI=”[some URI]”>[some name]</agent>. -->
          </maintenanceEvent>
        </maintenanceHistory>
      </control>
      <cpfDescription>
        <identity>
          <entityType value="person"/>
          <nameEntry>
            <part></part>
            <!-- Needs to be there, but can be left empty. Contrary to <agencyName> and <agent>, <part> itself
            cannot have @valueURI, but <nameEntry> can. So there might again be the question if we would allow
            the URI in <nameEntry> to replace the literal in <part>, i.e. having something like <nameEntry
            valueURI=”[some URI]”><part/></nameEntry>, or rather would highly recommend to have a
            literal nonetheless,  e.g. for the purpose of display and activation of clickable links, i.e. having
            something like <nameEntry valueURI=”[some URI]”><part>[some name]</part></nameEntry>.
          </nameEntry>
        </identity>
      </cpfDescription>
    </eac>
```

## Questions and options

### *@valueURI instead of literal*

Should it be possible to replace the literal in an element by only providing a @valueURI to the entity “named” in the element? This is a question for <agencyName>, <agent>, and <part> in connection with its parent element <nameEntry>.

- Option 1: Yes.
    - Follow-up question 1: Should we then - if technically/easily possible via the schema definition - require for EITHER @valueURI to be present and to not be empty OR for the element to have content?
    - Follow-up question 2: Should we then allow for <part> within <nameEntry> to be optional, so that instead of <nameEntry valueURI=”[some URI]”><part>[some name]</part></nameEntry> we could simply have <nameEntry valueURI=”[some URI]”/>?
- Option 2: No, @valueURI should only be used in connection to a literal being provided in the according element.
    - Follow-up question 1: Should we make this into a general recommendation for all elements that can have @valueURI, i.e. not only the ones that are mandatory in a minimal EAS file?
    - Follow-up question 2: Should we then require <agencyName>, <agent>, and <part> within <nameEntry> to have content?

- Alternative option: Is there any of these elements, <agencyName>, <agent>, and <part> within <nameEntry>, that could be made optional?

*@standardDateTime instead of literal*

Should it be possible to replace the literal in <eventDateTime> by only providing a @standardDateTime?

- Option 1: Yes.
  - Follow-up question 1: Should we then - if technically/easily possible via the schema definition - require for EITHER @standardDateTime to be present and to not be empty OR for the element <eventDateTime> to have content?
  - Follow-up question 2: Or should we go one step further and use a similar model for <eventDateTime> as we have for <entityType>, i.e. requiring the attribute @standardDateTime along with keeping <eventDateTime> mandatory, but not allowing for content in the element anymore?
- Option 2: No, @standardDateTime should only be used in connection to a literal being provided in the element <eventDateTime>.
  - Follow-up question: Should we then require <eventDateTime> to have content?
  - Alternative option: Could <eventDateTime> be made optional?

## Current EAD3 (deprecated)

```
<ead xmlns="http://ead3.archivists.org/schema/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ead3.archivists.org/schema/ https://www.loc.gov/ead/ead3.xsd">
  <control>
    <recordid></recordid>
    <!-- Needs to be there, but can currently be left empty. It would, however, make sense, if EAD followed
    EAC-CPF in this as part of the major revision. -->
    <filedesc>
      <titlestmt>
        <titleproper></titleproper>
        <!-- Needs to be there, but can be left empty. -->
      </titlestmt>
    </filedesc>
    <maintenancestatus value="new"></maintenancestatus>
    <!-- Needs to be there, but can be left empty. As EAD will follow EAC-CPF in this as part of the major
    revision, <maintenancestatus> will become a mandatory attribute of <control>. -->
    <maintenanceagency>
      <agencyname></agencyname>
      <!-- Needs to be there, but can be left empty. Same solution as decided for EAC-CPF above should be
      applied in EAD, too, as part of the major revision. -->
    </maintenanceagency>
    <maintenancehistory>
      <maintenanceevent>
        <eventtype value="created"></eventtype>
        <!-- Needs to be there, but can be left empty. As EAD will follow EAC-CPF in this as part of the major
        revision, <eventtype> will become a mandatory attribute of <maintenanceEvent>. -->
        <eventdatetime></eventdatetime>
        <!-- Needs to be there, but can be left empty. Same solution as decided for EAC-CPF above should be
        applied in EAD, too, as part of the major revision. -->
        <agenttype value="human"></agenttype>
        <!-- Needs to be there, but can be left empty. As EAD will follow EAC-CPF in this as part of the major
        revision, <agenttype> will become a mandatory attribute of <agent>. →
```

```
            <agent></agent>
            <!-- Needs to be there, but can be left empty. Same solution as decided for EAC-CPF above should be
            applied in EAD, too, as part of the major revision. -->
          </maintenanceevent>
        </maintenancehistory>
      </control>
      <archdesc level="fonds">
        <did>
          <unittitle></unittitle>
          <!-- At least one sub-element of <did> needs to be present, i.e. one of <abstract>, <container>, <dao>,
          <daoset>, <didnote>, <langmaterial>, <materialspec>, <origination>, <physdescset>, <physdesc>,
          <physdescstructured>, <physloc>, <repository>, <unitdate>, <unitdatestructured>, <unitid>, <unittitle>.
           -->
        </did>
      </archdesc>
    </ead>
```

### Questions and options

In addition to the elements shared with EAC-CPF - and hence already described above, EAD3 includes two more cases where mandatory elements can be left empty at the moment. These do not have to be decided now, as they will be part of the major revision of EAD any way, but it might be good for the Schema Team to already think about these in the context of the above and to provide the EAD Team with an indication of possible direction.

#### *<titleproper> within <titlestmt>*

Should we require <titleproper> within <titlestmt> to have content?

- Option 1: Yes.
    - Additional comment: We could potentially recommend that someone who does not have a dedicated “finding aid title” could simply repeat the <unittitle> of the highest <c> (or <c01>) element for <titleproper>.
- Option 2: No.
    - Follow-up option A: Keep it as is and require <titleproper>, but allow it to be kept empty.
    - Follow-up option B: Make <filedesc> and thereby <titlestmt> and <titleproper> optional.

#### *Sub-elements of <did>*

Is there one of the sub-elements of <did> that could be made mandatory (and repeatable) specifically rather than requiring any of the sub-elements of <did> to be present?

- Option 1: Following ISAD(G) - and now RiC-CM - maybe <unittitle> could be an option?
    - Follow-up question: If <unittitle> would be made a required sub-element of <did>, should we require <unittitle> to have content?
- Option 2: Following ISAD(G) - and now RiC-CM - maybe <unitid> could be an option?
    - Follow-up question: If <unittitle> would be made a required sub-element of <did>, should we require <unittitle> to have content?